



TITLE:

大島スプライン曲線の応用と KeTCindyプログラミング (数学ソ フトウェアとその効果的教育利用 に関する研究)

AUTHOR(S):

高遠, 節夫

CITATION:

高遠, 節夫. 大島スプライン曲線の応用とKeTCindyプログラミング (数
学ソフトウェアとその効果的教育利用に関する研究). 数理解析研究所
講究録 2019, 2105: 126-136

ISSUE DATE:

2019-02

URL:

<http://hdl.handle.net/2433/251882>

RIGHT:

大島スプライン曲線の応用と KeTCindy プログラミング

東邦大学・理学部 高遠 節夫

Setsuo Takato, Faculty of Sciences, Toho University

1 はじめに

Cinderella ([1]) は動的幾何ソフト (DGS) の 1 つであり, 他の DGS と同様に図形表示の画面 (CindyScreen) 上に幾何要素を描き, 動かすことができる. さらに, CindyScript というプログラミング言語をもっている. これは, 幾何要素の操作にとどまらず, 一般のプログラミング言語のように, いろいろな要素を扱う関数を作ることができる. 実際, 私たちが開発している KeTCindy は CindyScript のマクロパッケージであり, L^AT_EX をベースとし, 様々な図を教材や研究論文に挿入するための教材作成支援ツールである. ([8],[9]). CindyScreen と CindyScript エディタの画面は図 1 のようになっている.

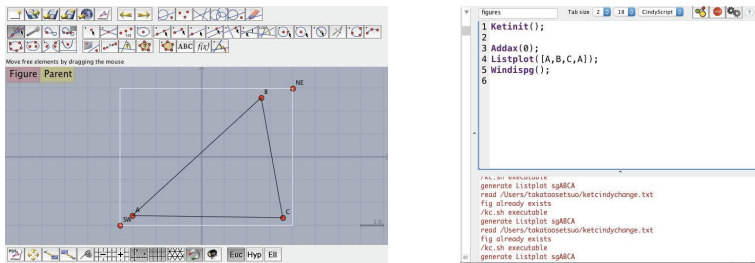


図 1 CindyScreen と CindyScript エディタ

数学の教員は, 印刷教材を作成して授業で配付することが多い. その際, 大学初年級・高専では L^AT_EX がしばしば用いられるが, それは, 彼らは普段から論文を作成するために L^AT_EX を使うことが理由と考えられる. KeTCindy を用いれば, それらの教材に必要な図を容易に作成することができる. KeTCindy は, ketpic.com ([4]) のトップページにある Link List から OS ごとの Installer に行けば, 無償でダウンロードすることができるになっているが, 最近では CTAN (Comprehensive TEX Archive Network) [5] からもダウンロード可能である. すなわち, CTAN のページに行って, ketcindy でキーワード検索すればよい. ただし, KeTCindy のパッケージ以外に, 必須のソフトとして Cinderella [1], R [2] および数式処理を利用するのであれば Maxima [3] をインストールする必要がある.

一例として, ここでは自由曲線を用いた教材の作成方法を示すことにする. 自由曲線は数学の教材作成にしばしば有用であるが, KeTCindy は, Bezier, Mkbezierptcrv, CRspline, Ospline など, Bézier 曲線に関するいくつかのコマンドを実装している. これらのうち, Bezier と Mkbezierptcrv では, 通過点 (節点) とそれ以外の制御点をすべて与える必要があるが, CRspline と Ospline は通過点を与えるだけでスムーズな曲

線を作るものである．ここでは，Catmul-Rom スプラインを描く `CRspline` を用いることにする．`KETCindy` で \LaTeX 用図データ（以下，プロットデータ）を作成する手順は次の通りである．

1. `KETCindy` をインストールしてできた作業ディレクトリにある `template` またはサンプルファイルをダブルクリックする．このとき画面上に現れる白い枠は \LaTeX の `picture` 環境の描画領域を表している．
2. 画面上の点 `SW` と `NE` を動かせば，描画領域を変えることができる（図2）．さらに，画面上に点 `A`, `B`, `C`, `D` を追加する（図3）．これらの点も動かすことができる．

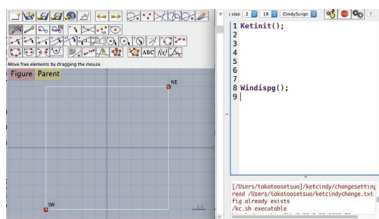


図2 初期画面

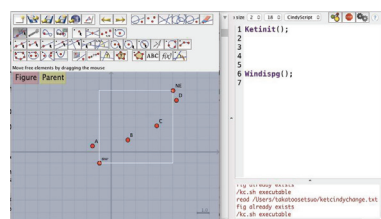


図3 点の追加

3. `CindyScript` エディタにコマンド `CRspline("1",[A,B,C,D]);` を書き込むと自由曲線が画面上に現れる（図4）．画面上で点 `A`, `B`, `C`, `D` を動かして曲線の形状を変えることができる．
4. `CindyScreen` の左上の `Figure` ボタンを押すと，`R`， \LaTeX および `PDF` ビュアーが順に実行されて，図5のプロットデータが作成表示される．

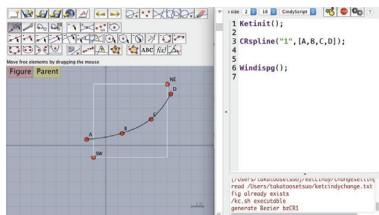


図4 CR スプラインの描画

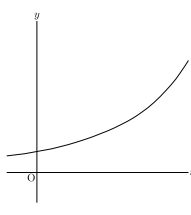


図5 プロットデータの表示

5. 図に追加したい他の要素を作成するコマンドを，例えば次のように書き込む．

```
CRspline("1",[A,B,C,D]);
pt=[C.x,B.y];
Listplot("1",[B,pt,C,B]);
Bowdata("1",[B,pt],["do","E=\Delta x"]);
Bowdata("2",[pt,C],[1.5,"do","E=\Delta y"]);
Letter(B,'n','P');
Htickmark([B.x,"x",C.x,"xi"]);
Vtickmark([B.y,"f(x)",C.y,"f(xi)"]);
```

再び Figure ボタンを押せば、教材用のプロットデータが完成する。

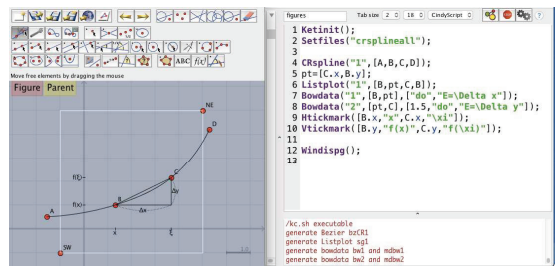


図 6 他の要素の追加

効果的な印刷教材のためには、図や文などの要素を適切に配置することが必要である。私たちが開発した ketpic.sty, ketlayer.sty には、layer という TeX の環境が定義されていて、教材を構成する要素（特に図）を教材の中に自由に配置することができる。

```
\begin{layer}{150}{0}
\putnotese{80}{5}{\input{fig/graph1.tex}}
\end{layer}
```

図 7 はこのようにして作成された図を用いた教材例である。

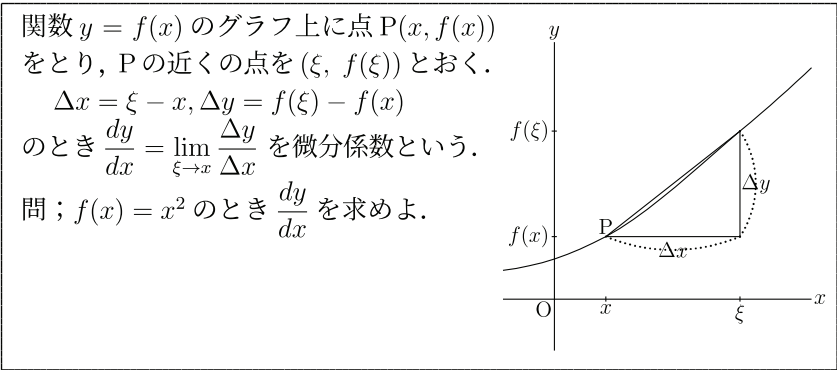


図 7 教材例

Catmul-Rom スプライン曲線はよく知られており、自由曲線を描いたり非線形補間を行うときにしばしば用いられている。しかし、その補間が有効でない場合も少なくない。例えば、図 8 のように、楕円上の 4 点 A, B, C, D で作られる Catmul-Rom スプライン曲線は楕円から大きくずれている。

```
Paramplot("1", "[3*cos(t), 2*sin(t)]", "t=[0, 2*pi]", ["da"]);
// to show the ellipse by dashed line.
A=[3,0]; B=[0,2]; C=[-3,2]; D=[0,-2];
CRspline("1", [A,B,C,D,A]);
```

ここで、黒点は Catmul-Rom スプライン曲線の制御点である。

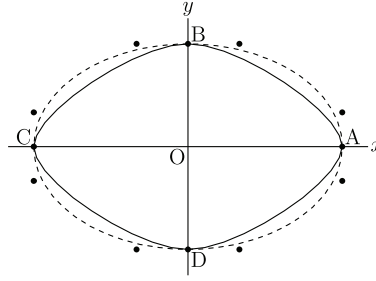


図 8 楕円を近似する Catmul-Rom スプライン曲線

そこで、大島利雄氏は制御点のとり方を工夫して、より精度の高い近似を与えるスプライン曲線を求める方法を提案した ([6])。KETCindy では、このスプライン曲線を描くコマンド `Ospline` を実装した。次節では、その詳細と応用について述べることにする。

2 大島スプライン曲線

P_{j-1} , P_j , P_{j+1} , P_{j+2} は曲線の通過点（節点）とし、区間 $P_j P_{j+1}$ の制御点を Q_j , R_j とおく。Catmul-Rom スプライン曲線の場合は、これらの制御点は

$$\overrightarrow{P_j Q_j} = \frac{1}{6} \overrightarrow{P_{j-1} P_{j+1}}, \quad \overrightarrow{P_{j+1} R_j} = \frac{1}{6} \overrightarrow{P_{j+2} P_j}$$

によって定められる。上の方程式で係数は定数 $\frac{1}{6}$ であることに注意する。このため、図 8 の点 A, B の近くで見られるように、曲線の曲がり具合の大きい部分が現れる。これを改良するために、大島氏は次の方程式を提唱した。

$$\overrightarrow{P_j Q_j} = c \overrightarrow{P_{j-1} P_{j+1}}, \quad \overrightarrow{P_{j+1} R_j} = c \overrightarrow{P_{j+2} P_j}, \quad \frac{P_j Q_j + R_j P_{j+1}}{P_j P_{j+1}} = \frac{4}{3(1 + \cos \frac{\theta}{2})}$$

ただし、 θ は線分 $P_{j-1} P_{j+1}$ と $P_j P_{j+2}$ の間の角である。

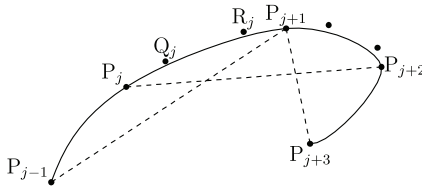


図 9 Catmul-Rom スプライン

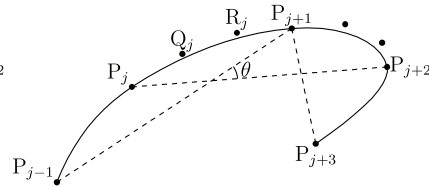


図 10 大島スプライン

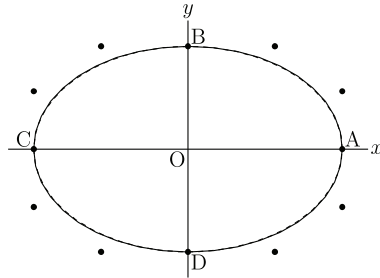


図 11 楕円と大島スプライン曲線

図 9 と図 10 はそれぞれ Catmul-Rom スプライン曲線と大島スプライン曲線を同一の節点と制御点で描いたものであり、後者の曲線が全体としてふっくらしているのが見られる。また、図 11 は先の例の楕円（図 8 の破線）について大島スプライン曲線を重ねたものであり、2つの曲線はほとんど重なっている。

大島スプライン曲線が常に最良の非線形補間を与えるとは言えないにしろ、多くの場合において、Catmul-Rom スプライン曲線に比してより良好な近似が得られる。私たちは大島スプライン曲線を用いて、数値積分 Integrate と数値微分 Derivative、Findarea のアルゴリズムを改良した。なお、大島氏は、複素積分を含む数値積分および不定積分の計算ライブラリを Risa/Asir で開発し、公開している ([7])。

Integrate

点 $P_j, P_{j+1}, P_{j+2}, P_{j+3}$ で定まる Bézier 曲線を C とし、各 P_k の座標を (x_k, y_k) とおく。 C の方程式は次のようになる。

$$P = P_j(1 - t)^3 + 3P_{j+1}(1 - t)^2t + 3P_{j+2}(1 - t)t^2 + P_{j+3}t^3 \quad (0 \leq t \leq 1)$$

積分 $\int_{x_j}^{x_{j+3}} y \, dx$ は次で計算される。

$$\int_{x_j}^{x_{j+3}} y \, dx = \int_0^1 y \frac{dx}{dt} dt \quad \text{ただし } P(x, y) \tag{1}$$

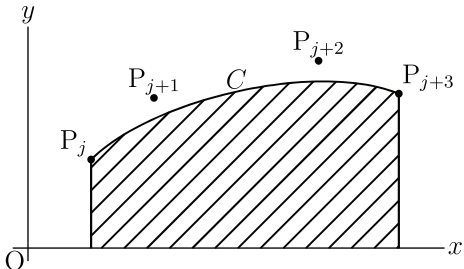


図 12 Bézier 曲線の積分

KETCindy は Maxima を呼んで数式の計算をし、結果を文字列として受け取るコマンドを実装している ([9])。これらのうち、Mxfun および CalcbyM は、それぞれ単独および複数の Maxima コマンド用であり、上の (1) は CalcbyM を用いて次のように計算される。

```
cmdL=[
  "P:[x1,y1]*(1-t)^3+3*[x2,y2]*(1-t)^2*t
    +3*[x3,y3]*(1-t)*t^2+[x4,y4]*t^3", [],
  "f:P[2]*diff(P[1],t)", [],
  "ans:integrate",["f","t",0,1],
```

```
"ans", []
];
CalcbyM("ans",cmdL);
println(ans);
```

結果は以下の文字列として得られる.

$$\begin{aligned} & ((10x^4 - 6x^3 - 3x^2 - x)y^4 + (6x^4 - 3x^2 - 3x)y^3 \\ & + (3x^4 + 3x^3 - 6x)y^2 + (x^4 + 3x^3 + 6x^2 - 10x)y)/20 \end{aligned}$$

この式を組み込んで修正した `integrate` の計算例をいくつか示すことにする.

例 1 $\int_0^{\pi} x^2 \sin x \, dx$

```
Plotdata("1", "x^2*sin(x)", "x=[-pi,pi]", ["Num=50"]);
ans=Integrate("gr1", [0,pi]);
println(Sprintf(ans,6));
```

結果は 5.869063 であり, Maxima によって確認することができる.

```
Mxfun("1", "integrate", ["x^2*sin(x)", "x", 0, "%pi"]);
Mxfun("2", "float", [mx1]);
```

Maxima による計算結果は $\pi^2/4$ で, 浮動小数点では 5.869604401089358 である.

例 2 楕円 $\frac{x^2}{3^2} + \frac{y^2}{2^2} = 1$ で囲まれた領域の面積

`Findarea` は, 閉曲線で囲まれた領域の面積を求めるコマンドである.

```
Paramplot("1", "[3*cos(t),2*sin(t)]", "t=[0,2*pi]", ["Num=50"]);
ans=Findarea("gp1");
println(Sprintf(ans/(6*pi),6));
```

結果は 0.999998 であり, $\frac{S}{\pi ab}$ の良好な近似を与えている.

注 上の例に大島氏のライブラリ ([7]) を適用すると

```
eval(os_md.areabezier([-3*cos(x),2*sin(x)],-50,[0,2*@pi]))/@pi/6);
```

で計算され, 結果は 0.999999828426758 とさらに精密になる.

`Integrate` は, 単に関数だけではなくプロットデータにも適用できることを注意しておく. 実際, 例 1 の `gr1` や例 2 の `gp1` は, 51 個の点からなるリストである.

例 3 陰関数の積分

`KETCindy` のコマンド `Implicitplot` によって, 陰関数で定義される関数のデータが得られる. そのときの積分は次のスクリプトで求められる.

```
Implicitplot("1", "8*x^2-4*sqrt(2)*x*y+y^2-3*x-6*sqrt(2)*y+2=0",
            "x=[-2,2]", "y=[-2,2.5]");
P=Ptstart("imp1");
Q=Ptend("imp1");
Letter([P,"n","P",Q,"n","Q"]);
ans=Integrate("imp1", [Q_1,P_1]);
println(Sprintf(ans,6));
```

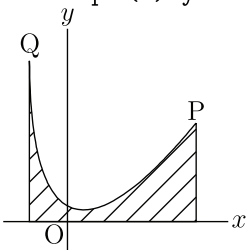


図 13 陰関数の場合の積分

結果は 1.699186 である。

Derivative

プロットデータを大島スプライン曲線で近似することによって、その数値微分を求めることもできる。図 12 の点 P_j における微分は、式 (1) を t で微分して $t = 0$ とおけば

$$\frac{dP}{dt} = -3P_j + P_{j+1}$$

これを用いて、コマンド Derivative およびプロットデータの接線を描く Tangentplot を実装した。以下は、例 3 の曲線に Tangentplot を適用して接線を引く例である。

```
dx=(Q_1-P_1)/10;
forall(0..10,ii,
    name=text(ii);
    v=P_1+ii*dx;
    rng="x="+format(v,6);
    Tangentplot(name,"imp1",rng,["dr,0.2"]);
);
```

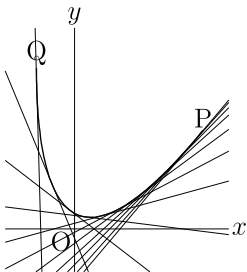


図 14 陰関数の場合の接線

3 曲面の描画への応用

数学の授業では、空間曲面を入れた教材がしばしば必要となる。その際、印刷教材には線画が適している。線画であれば、学生自身が式や補助線を書き入れることができるからである (図 15)。KE₂TCindy はこのような線描画をサポートしている。

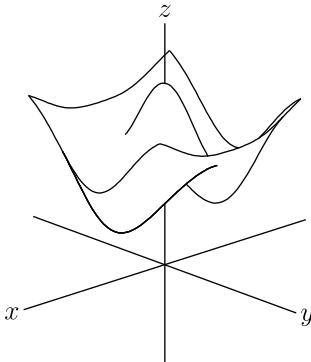


図 15 曲面の線描画

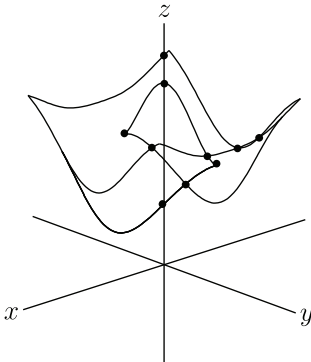


図 16 曲線の分割

KE TCindy の線画の作成手順は以下の通りである.

Step 1) まず, 曲面 $x = x(u, v)$, $y = y(u, v)$, $z = z(u, v)$ の輪郭線 (シルエットライン) を次の陰関数から求める.

$$J(u, v) = \frac{dX}{du} \frac{dY}{dv} - \frac{dX}{dv} \frac{dY}{du} = 0$$

ここで, $(X, Y) = \text{Proj}(x, y, z)$ は投射面への射影である.

Step 2) 次に, 輪郭線および境界線と射影された平面曲線との交点を求める (図 16).

Step 3) これらの交点によって平面曲線を分割し, 各部分が曲面で隠されているかどうかを決定する.

上に挙げた手順のうち, 陰線処理のためには Step 2 が本質的であるが, 曲線どうしが接するような場合は, プロットデータの実態は多角形であるため, 交点を求めることが難しくなる (図 17 参照).

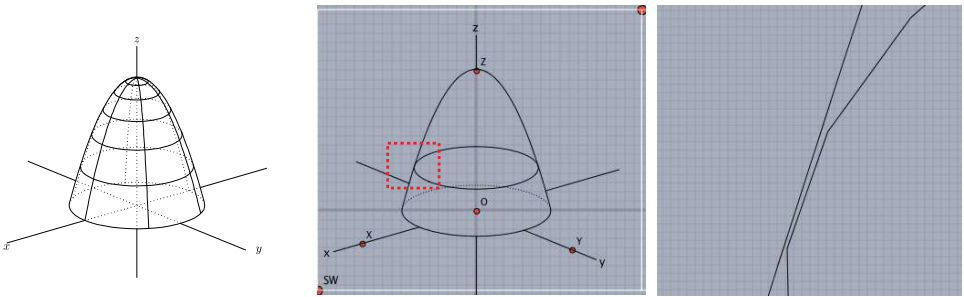


図 17 曲線が接する場合

そこで, Step 2 の計算を精密化するために, 接点の近くの曲線部分 (線分) を大島スプライン曲線によって補間することにし, そのアルゴリズムを平面曲線の交点を求めるコマンド `Intersectcurves` に組み込むことにした (図 18).

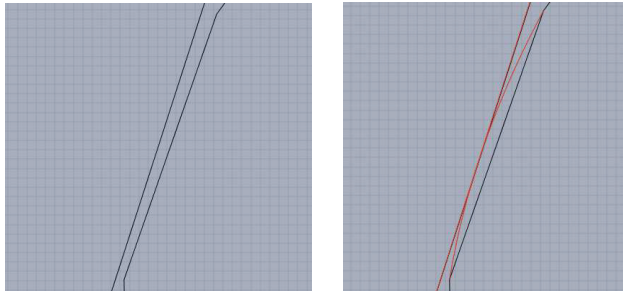


図 18 大島スプライン曲線の利用

この結果、交点を求め損なう場合は大幅に減少し、陰線処理の精度は大きく改善された。接点の近くでは、一般には近隣にある複数の点が返されることになるが、上記の円錐の例では、求められた交点の1つは次のようである。

$$P = [-1.65827, 1.20578] \quad (1)$$

この例については、Maxima で正確な交点を求めることができる。結果は

$$P = [-1.656701299244927, 1.210755779027779] \quad (2)$$

であり、(1) が良好な近似を与えていることが見られる。

さらに、陰線処理にかかる時間を短縮するために、K_ETCindy から C コンパイラ gcc を呼ぶコマンドを実装していたが、それを改良した。gcc によって図 17 左図の T_EX ファイルを生成する CindyScript のスクリプトは以下の通りであり、CindyScript や R では数分かかっていたのが、数秒程度で図が作成されるようになった。

```
fd=[
  "z=4-(x^2+y^2)",
  "x=R*cos(T)", "y=R*sin(T)",
  "R=[0,2]", "T=[0,2*pi]", "e"
];
Startsurf();
Sfbdparadata("1",fd);
Crvsfparadata("1","ax3d","sfbd3d1",fd);
Wireparadata("1","sfbd3d1",fd,5,2*pi/6*(0..5));
ExeccmdC("1",[""],[]);
```

K_ETCindy はパラパラ動画や animate.sty によるアニメーションを作成する機能を持っており、gcc による高速化は空間曲面の動画を作成することをも容易にした。ここでは、Möbius の帯の例を紹介することにする。まず、動画を作るときの CindyScreen は次の図 19 のようになっている。

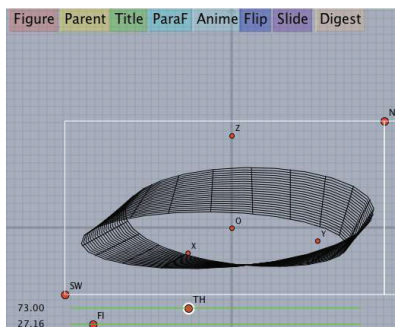


図 19 空間図形の CindyScreen

また、以下のスクリプトを CindyScript エディタに書き込む。

```
Start3d([A,B,C,S,S1,Sr]);
mf(s):=(
  regional(tmp);
  Startsurf();
  fd=["p",
    "x=2*cos(t)*(2+r*cos(t/2))",
    "y=2*sin(t)*(2+r*cos(t/2))",
    "z=2*r*sin(t/2)", "r=[-0.4,0.4]", "t=[0,"+text(s)+"]", "nsew"
  ];
  Sfbdparadata("1",fd);
  tmp=select((1..12)/12*2*pi,#<=s);
  Wireparadata("1","sfbd3d1",fd,5,tmp);
  ExeccmdC("",[""],["nodisp"]);
);
Setpara("mobius","mf(s)","s=[0,2*pi]",["Div=24"]);
```

パラパラ動画では Flip ボタン、アニメーションでは Anime ボタンを押せば、動画を構成する各スライドが生成される。

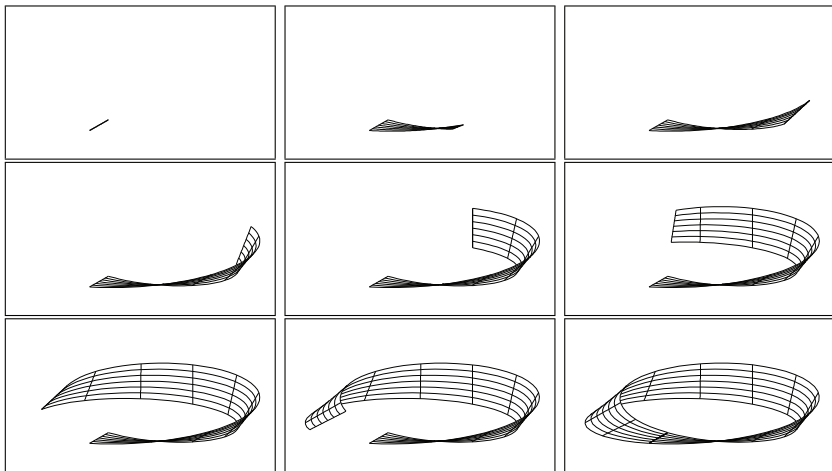


図 20 Möbius の帯のスライド

4 まとめ

大島スプライン曲線は、多くの場合において非線形補間の精度が高く、本稿で取り上げたようにいろいろな応用が考えられる。教員が数学ソフトウェアを用いて教材を作ろうとすると、そのソフトウェアが提供する機能だけでは足りない場合も多い。そのた

めに普通の教員が容易に使えるプログラミング言語は重要な要素であり, CindyScript はそのような言語の 1 つである. 加えて, 図 17 や図 18 のように, 画面上で確認しながらプログラミングを進めることができる点も大きな長所である. KeTCindy は, CindyScript を用いることで, 多くの機能拡張がなされている. また, KeTCindy から R, Maxima, C などの呼び出しができることは, より適切な教材を作るために有用である場面も少なくない. 現時点では, ベジェ曲線どうしの交点を数値的な方法により求めることで接点の近似値を得ているが, 大島氏から, さらに工夫の余地があるのではないかと助言をいただいた. これについては C との整合性の検討と合わせて, 今後の課題としたい.

謝辞 本研究は JSPS 科研費 16K0115, 18K02948, 18K02872 の助成を受けている.

参考文献

- [1] Cinderella, <http://www.cinderella.de/tiki-index.php>
- [2] R, <https://www.r-project.org>
- [3] Maxima, <http://maxima.sourceforge.net>
- [4] KeTpic, <http://ketpic.com>
- [5] CTAN, <https://ctan.org>
- [6] Oshima T., Drawing Curves, Mathematical Progress in Expressive Image Synthesis III, edited by Y. Dobashi and H. Ochiai, Mathematics for Industry, 24, 95–106, Springer, 2016 ISBN : 9789811010750.
- [7] Oshima, T., os muldif.rr, a library for a computer algebra Risa/Asir, 2008-2018, <ftp://akagi.ms.u-tokyo.ac.jp/pub/math/muldif/>
- [8] Takato S., What is and how to Use KeTCindy – Linkage Between Dynamic Geometry Software and Collaborative Use of KetCindy and Free Computer Algebra Systems and L^AT_EX Graphics Capabilities –, Mathematical Software –ICMS 2016, LNCS **9725**, 371–379, Springer, 2016.
- [9] Takato S, McAndrew, Vallejo J, Kaneko M., Collaborative use of KeTCindy and free Computer Algebra Systems, Mathematics in Computer Science 11 3-4 , 503-514, 2017.
- [10] Takato S, Vallejo J, Interfacing Free Computer Algebra Systems and C with KeTCindy, Computer Algebra Systems in Teaching and Research, Siedlce University of Natural Sciences and Humanities Volume 6, 172–185, 2017
- [11] Takato S, Vallejo J, Hamiltonian dynamical systems: symbolical, numerical and graphical study, to appear in Mathematics in Computer Science, 2018.